

综合维度学习的多群协作粒子群优化算法^{*}

张其文, 王杨婷[†]

(兰州理工大学 计算机与通信学院, 兰州 730050)

摘要: 针对维度学习策略(Dimensional learning strategy, DLS)中存在的“过度开发”问题, 提出了一种综合维度学习的多群协作粒子群优化算法(CDL-MCP SO), 为提高种群搜索效率, 算法采用基于主从范式的集群结构, 将种群划分为一个主群和四个从群, 主群执行综合学习策略在搜索空间进行大范围探索, 从群执行综合维度学习策略(Comprehensive Dimensional Learning, CDL)在局部最优解附近进行高精度地开发, 主从群通过执行具有不同职能的算法能够有效实现其在勘探和开发之间的平衡, 同时为保持种群多样性, 提出了一种新的解交换机制(SEM), 用来在主从群独立运行各自算法若干代之后进行信息的交流与协作, 以指导粒子后期进行更准确的搜索, 最后, 针对初始化过程随机性过高, 运用拉丁超立方体采样方法对算法重建输入分布, 为验证 CDL-MCP SO 的有效性, 将其与 5 种粒子群算法变体在 10 个测试函数进行实验对比, 结果表明该算法总是可以找到优于或相当于对比算法的解, 在求解复杂函数时具有可行性和高效性。

关键词: 粒子群算法(PSO); 综合维度学习策略; 主从范式; 多群协作

中图分类号: TP301.6 **doi:** 10.19734/j.issn.1001-3695.2022.01.0019

Multi-swarm collaborative particle swarm optimization algorithm based on comprehensive dimensional learning

Zhang Qiwen, Wang Yangting[†]

(School of Computer & Communication, Lanzhou University of Technology, Lanzhou 730050, China)

Abstract: Aiming at the problem of "over-exploitation" in dimensional learning strategy (DLS), this paper proposed a multi-swarm collaborative particle swarm optimization algorithm with comprehensive dimension learning (CDL-MCP SO). The cluster structure of master-slave paradigm divides the population into a master group and four slave groups, in which the master group executes a comprehensive learning strategy to conduct a large-scale exploration in the search space, and the slave groups execute a comprehensive dimensional learning strategy (CDL) to exploit high-precision solutions near the local optimal. The master-slave groups can effectively achieve the balance between exploration and exploitation by executing algorithms with different functions. At the same time, in order to maintain the diversity of the population, the CDL-MCP SO proposed a new solution exchange mechanism(SEM). By doing this, the algorithm can accomplish information exchange and cooperation after the master-slave groups running their respective algorithms for several generations independently, so as to guide the particles to conduct more accurate searches in the later stage. Finally, for the high randomness of the initialization process, the algorithm adopted latin hypercube sampling to reconstruct the input distribution. In order to verify the effectiveness of CDL-MCP SO, compared with 5 kinds of PSO variants in 10 test functions. the results show that the algorithm can always find better or equivalent solutions. It is feasible and efficient in solving complex functions.

Key words: particle swarm algorithm(PSO); comprehensive dimensional learning strategy; master-slave paradigm; multi-group collaborative

0 引言

近年来, 智能优化算法逐渐成为解决科研和工程应用领域中许多复杂优化问题的首选方法。常见的主要包括粒子群算法(PSO)、和声搜索算法(HS)、文化算法(CA)、基于教与学的算法(TLBO)以及重力搜索算法(GSA)等。其中最经典的当属模拟生物行为机制的粒子群优化算法(particle swarm optimization, PSO)^[1], 它是由 Kennedy 和 Eberhart 博士于 1995 年正式提出的一种基于种群的元启发式优化算法, 由于其实现简单、控制参数少吸引了大量学者研究并涌现出了许多变体。针对它的改进主要有四个方面: 参数设置、拓扑结构、算法融合以及学习策略。

参数的取值对于平衡算法的勘探和开发能力有着重要的作用。Shi 等人^[2]通过研究发现粒子对先前速度的继承能力对

算法性能的提升有很大作用, 进而在基本粒子群算法中引入了惯性权重, Sarhani 等人^[3]为平衡算法的全局与局部搜索能力, 提出一种基于隐性马尔可夫模型状态估计的惯性权重控制机制, 实验结果证明了算法的收敛优势。2017 年, Gou 等人^[4]根据每个粒子之间的个体差异动态调整参数的值, 同时采用重启策略再生粒子, 实验结果证明了该算法具有鲁棒性和可扩展性。

在拓扑结构的选择方面, Nasir 等人^[5]提出了一种动态邻域学习粒子群优化算法(DNLPSO), 该算法从邻域(包括自身)的最佳位置选择一个样本粒子。文献[6]为了从根源上克服 PSO 算法易陷入局部最优的问题, 针对多种拓扑结构进行实验分析, 最终提出了一种全局探索与局部开发相融合的混合拓扑粒子群算法。

利用不同的进化算法来提高 PSO 的性能是研究人员的

收稿日期: 2022-01-22; 修回日期: 2022-03-11 基金项目: 国家自然科学基金资助项目(62063021, 62162040); 浙江省基础公益研究计划(LQ20F020011)

作者简介: 张其文(1975-), 男, 山西临汾人, 副教授, 硕导, 硕士, 主要研究方向为智能信息技术处理、知识发现、大数据等; 王杨婷(1996-), 女(通信作者), 陕西渭南人, 硕士研究生, 主要研究方向为智能优化算法、模式识别与人工智能(nanxun070610@163.com)。

又一重点。最早的算法融合是 Gong 等人^[7]将遗传算法中的三个经典操作选择、变异和交叉引入到 PSO 中。每个粒子根据遗传操作生成一个有希望的样本, 然后以标准 PSO 的方式从样本中学习。2018 年, Aydılek 等人^[8]将萤火虫算法与粒子群算法的概念进行融合, 实验结果证明了算法不仅能有效平衡算法的探索与开发能力, 而且运行时间和解的精度都有明显提升。Senel 等人^[9]将 PSO 的开发能力与灰狼优化算法的探索能力相结合, 在同一测试环境下, 该算法在短时间内获得较好的成效。实验结果表明算法能在更短的时间内找到更好的优化方案。

根据以上各种改进, 可以明确 PSO 变体的主要目标是保持种群勘探和开发能力的平衡, 并在保持种群多样性的同时提高算法的收敛速度以及解的质量, 但现实生活中大多数优化问题往往都比较复杂。在多样性方面, 单维问题需要较小的种群多样性, 以便快速收敛, 而多模态问题需要较好的种群多样性来避免算法陷入局部极值。由此可见只采用单一策略无法满足不同优化问题的需求。因此一些研究人员提出了分而治之的策略来解决复杂优化问题, 针对不同的优化目标运用的不同的学习策略。陈跃刚等人^[10]基于综合学习策略对多目标问题进行求解, 通过分解方法更新主导粒子以增强解的分布; 采用存档机制存储优化过程中的非支配解, 并采用多项式变异来避免陷入局部最优, Mendes 等人^[11]提出了完全知情粒子群优化算法(FIPS)。FIPS 利用完全联通领域中所有邻居的个人最佳信息的平均值更新粒子的位置, 文献[12]在考虑了种群结构、多模式学习和个体间博弈等因素之后, 提出了具有博弈概率选择的子种群粒子群算法。Parsopoulos 等人^[13]提出了统一粒子群优化算法(UPSO), 通过选择合理的邻域大小, 利用全局邻域和局部邻域的最佳信息来更新粒子位置, 袁小平等人^[14]将种群动态地划分为三个不同阶层, 并根据不同阶层粒子特性, 分别采用局部学习模型、标准学习模型以及全局学习模型, 来改善算法性能, 文献[15]开发了两种新的学习策略, 一方面, 提出了适应度景观中的局部稀疏度度量方法来估计粒子的拥塞和分布情况, 并在此基础上通过引导粒子到稀疏区域来建立探测策略。另一方面, 基于多群策略和自适应子群大小调整提出了一种自适应开发策略。Zhou 等人^[16]提出了一种自适应分层更新粒子群优化算法, 为两个群分别生成两层和三层更新公式, 并引入了多项选择综合学习策略。Şaban 等人^[17]基于综合学习和主从范式提出了并行综合学习粒子群优化算法(PCLPSO), 它基于主从范式, 并在不同范式之间进行解信息的协作。文献[18]受天体学和植物学启发, 提出一种多策略融合的粒子群优化算法。其中三黑洞系统捕获策略和多维随机干扰策略用来增强粒子的全局搜索能力, 而协调因子用来完成从全局寻优向局部搜索的转变, 以提高收敛速度。文献[19]根据 PSO 算法的收敛特性和 Logistic 映射的混沌思想, 提出一种粒子置换的双种群综合学习 PSO 算法(PP-CLPSO)。

从以上改进方法不难看出, 大多数针对参数、拓扑结构、算法融合的改进, 仅仅是针对其中一两个目标进行优化, 没有平衡好多个目标之间的关系, 多种策略分而治之的思想是根据算法的不同的优化目标设计不同的学习策略, 不仅能够实现勘探和开发的平衡, 还能有效避免早熟收敛、提高收敛速度以及解的质量。

CDL-MCP SO 和以上多策略融合的改进一致, 也采用了分而治之的思想, 但是考虑了绝大多数算法没有考虑到的“振荡”和“维度退化”现象, 例如文献[20]提出的维度学习策略, 虽然解决了“维度退化”现象, 但在不同群体中由于使用的学习策略并不完全适配子群职能而产生了新的问题-“开采过度”, 原因在于公式第二项的学习样本是全局最优, 在少数迭代之后, 会和公式第三项完全一致共同发挥开发作用, 开发

能力变成之前的两倍, 种群多样性逐渐丧失。针对此不足, CDL-MCP SO 通过融合综合学习策略(CLS)和维度学习策略(DLS)各自的优点, 提出“综合维度学习策略(CDL)”, 去掉公式第三部分对全局最优项的学习, 引入 CLS 中对所有其他粒子概率性学习的思想, 使粒子在开发的同时也可以适当探索, 并基于信息流动原理提出了一种新的解交换机制(SEM), 在提高算法收敛速度的同时也优化了解的质量, 最后通过拉丁超立方体采样对算法重建输入分布, 有效避免了初始化过程由于随机性过高给算法带来的负面影响。通过对改进思路产生原因的描述, 可以看出 CDL-MCP SO 具备新颖性和前沿性。

1 相关工作

1.1 粒子群优化算法

PSO^[2]源于对鸟群觅食共享个体知识社会行为的研究。群中的每个个体称为粒子, 代表搜索空间中的一个潜在解, 粒子通过跟随自身个体最优和种群全局最优来更新自己的速度和位置, 直到得到满足终止条件的最优解。粒子速度和位置更新公式如下:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_{1,j} (X_{i,j}^{pbest}(t) - X_{i,j}(t)) + c_2 r_{2,j} (X_j^{gbest}(t) - X_{i,j}(t)) \quad (1)$$

$$X_{i,j}(t+1) = V_{i,j}(t+1) + X_{i,j}(t) \quad (2)$$

其中: $i=1,2,\dots,m$, $j=1,2,\dots,n$, m 和 n 分别是种群规模和维数。 ω 指惯性权重, c_1 和 c_2 分别是自我认知和社会认知, $r_{1,j}$ 和 $r_{2,j}$ 是第 j 维上[0,1]之间均匀分布的两个随机数。 $X_{i,j}^{pbest}$ 和 X_j^{gbest} 分别代表粒子个体最优和全局最优。

1.2 综合学习策略(CLS)

在 2006 年 Liang 等人^[21]提出的 CLPSO 采用新的学习策略, 通过利用所有其他粒子的个体最优信息来更新速度, 为粒子提供了更多的学习标本和潜在的搜索空间, 在综合学习策略中, 粒子的更新由学习样本 $X_{i,j}^g(t)$ 来指导, 速度更新公式如下:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + cr(X_{i,j}^g(t) - X_{i,j}(t)) \quad (3)$$

位置更新见式(2), 其中 $X_{i,j}^g(t)$ 表示粒子 i 个体最优 $pbest$ 的第 j 维通过向所有其他粒子个体最优的第 j 维进行学习(包括粒子 i 自身个体最优的第 j 维)后产生的样本, 样本 $X_{i,j}^g(t)$ 学习过程: 粒子 i 个体最优每一维具体向谁学习取决于学习概率 p_c , 不同的粒子学习概率不同, 首先为每一维生成一个随机数 $rand$, 若随机数大于 p_c , 则向自己个体最优相应维度学习, 否则向其他粒子个体最优相应维度学习, 学习概率 p_c 更新公式如下:

$$p_c = 0.05 + 0.45 \times \frac{(\exp(\frac{10(i-1)}{ps-1}) - 1)}{\exp(10) - 1} \quad (4)$$

ps 代表种群规模, 在综合学习策略中, 其他粒子先前的最佳位置是任何粒子可以学习的样本, 并且粒子的每个维度都可以从不同的样本中学习, 这种融合来自不同学习样本信息的策略扩大了粒子的潜在搜索空间, 大大提高了种群的全局搜索能力, 因此在解决多模态问题效果优异, 但由于去掉了对全局最优项的学习, 导致在求解单峰问题上效果不佳。

1.3 维度学习策略(DLS)

Xu 等人^[20]针对 CLPSO 不足以及标准 PSO 在种群更新中存在的“振荡”^[22]和“两步走, 一步回头”^[23]问题, 提出了维度学习策略, 利用构造的学习样本 X_i^d 和全局最优 X^{gbest} 更新速度, 学习样本 X_i^d 的构造: 区别于综合学习策略中学习样本 X_i^g 的每一维来自其自身或所有其他粒子的相应维度。

DLS 学习样本 X_i^d 的每一维是来自其自身或全局最优 X^{gbest} 的相应维度, 学习与否取决于维度替换后粒子适应度值是否变好。这种策略可以将 X^{gbest} 的优秀信息传递给样本 X_i^d ,

促进了 X^{gbest} 编码模式的传播和保护。维度学习策略速度更新公式如下:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_{1,j} (X_{i,j}^d(t) - X_{i,j}(t)) + c_2 r_{2,j} (X_j^{gbest}(t) - X_{i,j}(t)) \quad (5)$$

位置更新按照标准粒子群更新式(2), 维度学习策略让个体最优 X_i^{pbest} 整合全局最优 X^{gbest} 有前途维度上的信息, 保证了学习后的样本不会比个体最优差, 同时将对全局最优 X^{gbest} 的学习项添加了回来, 不仅解决了“振荡”和“维度退化”现象, 也大大加快了算法的收敛速度, 但由于公式后两项在更新几代之后相差不大, 均侧重于开发, 极易出现“过度开发”。

2 综合维度的多群协作粒子群优化算法(CDL-MCPSO)

现有多群学习策略由于忽略了维度退化现象, 使得粒子位置在更新过程中会出现“前进两步, 后退一步”的问题, 即使粒子适应度变好了, 也只是因为在有些维度上粒子靠近全局最优的程度大于其他维度上粒子远离全局最优的程度, 导致粒子在靠近和远离全局最优之间反复横跳, 降低了算法搜索效率也影响了解的质量, 同时本文在对 DLS 速度更新公式演算之后发现, 公式第二项的 X_i^{pbest} 在向 X^{gbest} 学习几代后构造的学习样本 X_i^d 会慢慢地非常接近全局最优 X^{gbest} 的值, 此时对于后面粒子的更新, 公式后两项的学习样本均为全局最优, 赋予公式最后一项的开发能力由于后两项的无限接近变成了两倍, 而且粒子在更新过程中没有向其他粒子学习, 没有可以帮助粒子逃离“局部最优”的条件, 极易出现“开发过度”, 为解决此问题并提升算法运行效率, 本文首先采用主从范式的集群结构: 一个主群和四个从群, 其次在主从群中分别执行全局探索能力强的 CLS 和开发能力强的改进策略 CDL, 最后针对初始化随机性过高问题, 采用拉丁超立方体采样对算法重建输入分布, 在多群和适应子群职能更新规则双策略的加持下, 算法的搜索效率会大大提升, 同时从群更新规则的提出不仅可以解决“维度退化”问题, 还避免了“过度开发”。下面详细介绍 CDL-MCPSO 算法。

2.1 主从范式集群结构

为了提高算法勘探和开发之间的平衡, 采用基于主从范式的集群结构, 将整个种群划分为两类群体: 一个主群 (Master) 和四个从群 (Slave), 主群执行全局搜索能力强的综合学习策略在搜索空间内进行大范围的探索, 从群执行改进策略-综合维度学习策略在已经找到的较优解附近进行高精度的开发, 主从范式集群结构示意图如图 1 所示。

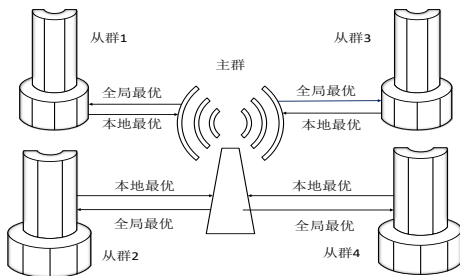


图 1 主从范式集群结构

Fig. 1 Master-slave paradigm cluster structure

2.2 主从群更新策略

为使从群能够在局部最优附近进行高精度的开发, 选择改进后的 DLS(CDL)作为其更新规则, 未被改进的 DLS 在文献[19]中被证明具有很强的开发能力, 但是它存在的“过度开发”问题, 会极易导致粒子陷入局部最优, 通过对其改进: 去掉 DLS 速度更新公式中对全局最优的学习项, 防止粒子个体最优 X_i^{pbest} 经过几代学习产生的样本 X_i^d 和第三项无限接近而导致快速向全局最优 X^{gbest} 靠拢, 加入 CLS 公式第二项粒

子个体最优对其他所有粒子个体最优的学习, 使得到的 CDL 可以适当削弱 DLS 的开发程度, 避免粒子过早陷入局部最优, 发挥出 DLS 最好的优化效果。速度更新公式具体如下:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_{1,j} (X_{i,j}^d(t) - X_{i,j}(t)) + c_2 r_{2,j} (X_{i,j}^d(t) - X_{i,j}(t)) \quad (6)$$

位置更新新式(2), 所提策略是对标准粒子群算法更新公式第二项和第三项换种思维的改进, 可以使算法在具有优秀开发能力的前提下适度探索, 由于 CDL 学习样本对粒子搜索方向的强大指引, 整体上开发能力更胜一筹, 因此在集群结构中用来对从群进行更新, 主群负责在大范围搜索空间进行探索, 因此选择全局搜索能力强的 CLS, 具体更新过程见 1.2 小节。

2.3 多群协作

主群在大范围搜索空间进行探索, 不仅能够得到较高质量的解, 而且还能很好地保持种群多样性, 但是会使算法的收敛速度变慢, 从群负责在较好解附近进行高精度的开发, 能够大大加快算法收敛速度, 但是若此时靠近的全局最优是局部最优, 会使粒子无法逃离此区域, 综上分析, 需要利用从群搜索到的信息来指导主群加快收敛速度, 也需利用主群搜索的信息来帮助从群粒子逃离“局部最优”。故主从群之间进行信息交流是必要的, 交流过程需要考虑以下四个问题: (1) 迁移周期; (2) 交流的对象; (3) 需要交换的信息; (4) 整合策略。为解决以上问题, 提出了一种新的解交换机制(SEM)。

(1) 迁移周期。表征的是主从群在独立执行各自算法多少代之后开始进行信息的交流与协作, 不管什么算法在优化一个具体问题的时候都要在解的精度和计算效率之间做折中, 如果信息交换的过于频繁, 算法运行时间会大大增加, 效率低下; 如果交换的间隔过于长, 又会使解的质量大打折扣, 在 SEM 中迁移周期的取值设置为 7。

(2) 要交流的对象。如果在太多群集之间交流, 会影响算法的运行效率, 但若交流的太少, 又会有很大的陷入“局部最优”的风险, 影响解的质量。同时由于从群主要负责在局部最优解附近进行高精度的开发, 执行的均为 DLS, 如果陷入局部最优, 利用具有勘探能力的主群搜索到的信息即可帮助其逃离, 所以从群之间无须进行交流, 如果交流反而会增加算法运行时间, 所以 SEM 只在主从群之间进行信息的交流。

(3) 需要交换的信息。群集之间通常交换的信息可以为全局最优、局部最优以及粒子邻居最优, 在 CDL-MCPSO 中, 主从群只在各自迭代时涉及对其他粒子的学习, 协作过程只是用来改善从群可能出现的“局部最优”和主群由于多样性强收敛速度不够快的情形, 因此无须交换邻居最优信息, 只需交换主群全局最优和各从群本地最优信息。

(4) 整合策略。处理交流后的信息, 用来指导粒子的后续搜索。整个解交换机制的详细过程总结如下, 主群更新策略: 在主从群独立执行各自的算法 7 代之后, 所有从群发送自己的本地最优 $lbest_i (i=1,2,3,4)$ 给主群, 主群根据(1)在所有从群本地最优 $lbest_i (i=1,2,3,4)$ 中最优的那个 $X_{gbest}^s (sbest)$ (2)主群自己的本地最优 X_{gbest}^M (3)主群粒子自身个体最优 X_{pbest}^M 对粒子进行更新, 这种更新策略是基于主群和从群之间的竞争来调整粒子运行轨迹。速度更新公式如下:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_{1,j} (X_{i,j}^d(t) - X_{i,j}(t)) + \varphi c_2 r_{2,j} (P_{gbest,j}^M(t) - X_{i,j}(t)) + (1-\varphi) c_3 r_{3,j} (P_{gbest,j}^s(t) - X_{i,j}(t)) \quad (7)$$

$$\varphi = \begin{cases} 0, Gbest^M < Gbest^s \\ 0.5, Gbest^M = Gbest^s \\ 1, Gbest^M > Gbest^s \end{cases} \quad (8)$$

其中: φ 是协作因子, 位置更新仍然采用式(2)。从群更新策略: 将所有从群全局最优 P_{gbest}^s 与主群全局最优 P_{gbest}^M 进行比较,

挑选出较优的那个替换在所有从群中随机选择出的一个粒子, 这样从群在间接进行了信息共享的同时也不会对算法的效率产生很大影响; 所提出的新的解交换机制不仅平衡了算法的勘探和开发, 还加快了算法的收敛速度, 同时解的质量也得到了很大的提升。表 1 和表 2 分别展示了解交换机制以及多群协作相关变量。

表 1 解交换机制

Tab. 1 Solution exchange mechanism

序号	具体操作
1	所有从群 S 发送自己的本地最优 $lbest_i (i=1,2,3,4)$ 给主群 M
2	计算 $Gbest^S = \min(f(lbest_i (i=1,2,3,4)))$
3	将 $X_{Pbest_i}^M$ 、 p_g^M 、 p_g^S 按照式(7)(2)对主群更新
4	计算 $Gbest = \min(p_{gbest}^M, p_{gbest}^S)$
5	在每个从群中随机选出一个粒子 $X_i (i=1,2,3,4)$
6	令 $X_i (i=1,2,3,4) = Gbest$

表 2 多群协作相关变量

Tab. 2 Multi-group collaboration related variables

变量	变量含义
M/S	主/从群
p_{gbest}^M	主群全局最优
p_{gbest}^S	主群全局最优
$X_{Pbest_i}^M$	主群粒子 i 个体最优
$lbest_i (i=1,2,3,4)$	从群本地最优
$Gbest^M$	主群全局最优适应度值
$Gbest^S$	从群全局最优适应度值
$Gbest$	群体全局最优

2.4 基于拉丁超立方采样的初始化策略

拉丁超立方体抽样最早由 McKay^[24]于 1979 年提出, 和普通抽样方法蒙特卡罗相比, 它通过对输入搜索空间进行等间隔划分, 并在每个区间中随机地抽取样本, 抽样被强制地代表每个区间的值, 具有均匀分布的特点, 能够保证在拥有较少样本的前提下具备比普通样本更好的高效性, 图 2 展示了在 2 维搜索空间中拉丁超立方体抽样示意图。通过拉丁超立方体抽样产生的样本能更好地均匀分布在空间, 可以有效避免算法出现早熟收敛的现象。具体采样步骤如下:

Step1: 确定抽样规模 H ;

Step2: 将每维变量 X_i 的定义域区间 $[X_{i1}, X_{in}]$ 划分成 H 个相等的小区间, 这样就一共有 H^n 个小超立方体产生;

Step3: 产生一个 $H \times n$ 的矩阵 A , A 的每列均是数列 $\{1, 2, \dots, n\}$ 的一个随机全排列;

Step4: A 的每行就只有一个超立方体被选中, 在每个小超立方体内产生一个样本, 这样就共有 H 个样本被选出, 且选出的样本互不相同。

图 2 和图 3 分别为拉丁超立方体抽样在二维空间的示意图以及所提算法的流程示意图。

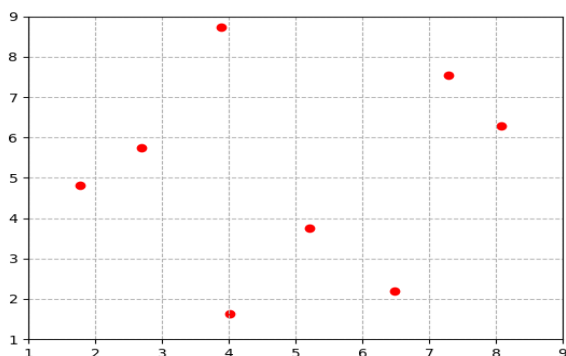


图 2 拉丁超立方体抽样(2 维)

Fig. 2 Latin hypercube sampling(2d)

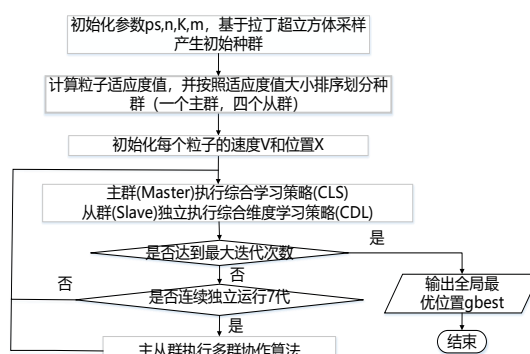


图 3 CDL-MCPSO 算法流程

Fig. 3 The algorithm flow of CDL-MCPSO

3 CDL-MCPSO 算法合理性分析

3.1 CLS 中去掉对全局最优项学习操作的合理性

由于 CDL 是基于 DLS 速度更新公式开发, 因此首先需要对 DLS 速度更新所采用策略存在的问题进行详细分析, 再阐述为何要进行 CDL 改进的原因, 具体分析如下: 和大多数 PSO 变体速度更新公式一样, DLS 速度更新公式也包含三个部分: 惯性部分 $\omega V_{i,j}(t)$ 、个体最优向全局最优学习构造学习样本, 向样本学习部分 $c_1 r_{1,j}(X_{i,j}^d(t) - X_{i,j}(t))$ 、向全局最优学习部分 $c_2 r_{2,j}(X_{jbest}^d(t) - X_{i,j}(t))$; 本文一般需要在公式后两项分别赋予种群探索和开发能力, DLS 更新公式最后一项对全局最优 $X_{jbest}^d(t)$ 的学习毫无疑问是赋予了种群开发能力, 那么在种群已经具备开发能力的前提下, 公式第二项个体最优 $X_{i,jbest}^d(t)$ 的学习对象仍然选择了会进一步增强种群开发能力的全局最优, 即使个体最优向全局最优学习后得到的新位置不是全局最优, 但是从其具体的学习过程来看, 它是逐维度进行学习, 每一次的学习并不一定只学习一个维度, 当得到的新位置的适应度值优于个体最优适应度值时学习过程才会停止, 这种学习过程即使放在高维问题中, 也会在若干代的学习之后使得公式第二项最终得到的学习样本 $X_{i,jbest}^d(t)$ 变成全局最优, 而不是介于个体最优和全局最优之间的一个值, 此时对于后续粒子的更新, 公式后两项将会完全一样, 赋予公式最后一项的开发能力由于后两项的接近甚至完全一致而变成两倍, 若此时粒子找到的全局最优是局部最优, 并且迭代过程中没有向其他粒子进行学习, 毫无疑问种群将面临陷入局部最优并且无法逃逸的风险, 需要说明的是, 公式第二项的学习样本在变成全局最优时经历的迭代次数和本文设置的最大迭代次数相比, 只会占一小部分比例。

针对以上分析存在的不足, 本文提出了 CDL 改进策略, 具体原因: 首先为削弱粒子在之后大部分迭代时期里拥有的两倍开发能力, 需要去掉其中一项(对全局最优的学习), 同时需要为种群引入增强探索能力的策略, 而针对多模态问题研发的 CLS 策略, 被证明在探索方面效果显著, 它能够在含有多个峰值的优化问题中找到最优解, 需要说明的是, 本文之所以去掉的是对全局最优的学习项, 而不是构造的学习样本项, 是因为在初始迭代过程中, 为了不遗漏任何可能包含最优解的期望区域, 本文希望算法在整体上是探索能力大于开发能力, 而留下来的学习样本项在初始阶段, 学习样本的值还不是全局最优, 因此对粒子的牵引力不算很明显, 种群中的粒子可以在搜索空间进行最大范围地探索, 在后期学习样本变为全局最优之后, 完全可以取代去掉的全局最优项来发挥其开发能力, 能够使算法在不同迭代时期最大程度地发挥勘探或开发能力, 综上所述 CDL 策略中去掉对全局最优项的学习是合理的。

3.2 CDL-MCP SO 克服过度开发问题的能力及其理论性依据

CDL-MCP SO 具备克服过度开发问题能力的理论性依据主要有三个: 1、基于主从范式的多群策略—能够弥补单个种群出现过度开发情形时, 没有其他种群信息指导的缺点; 2、从群的执行策略—在取消一半开发能力的前提下加入对其他粒子的学习, 不易出现过度开发; 3、解交换机制(SEM)的加持—即使开发过度陷入局部最优, 也可以通过信息的交流协作来帮助实现逃逸。具体说明如下:

基于主从范式的多子群相比单个种群具有更好的多样性, 能够减小粒子出现过度开发陷入局部最优的可能性, 因为可以利用其他种群搜索到的信息来进行指导, 其次是在每个子群运行算法的选择上, 让主群执行在多模态问题中优化效果显著的综合学习策略(CLS)以赋予主群进行大范围探索的能力, 显而易见, 主群不易出现过度开发的情形, 对于每个从群本文执行优化后的 DLS—CDL 着重赋予算法开发能力, 还加入了对所有其他粒子进行学习的部分, 更不会使算法出现过度开发的情形, 即使出现了, 新提出的解交换机制(SEM), 也可以通过信息的交换来增加种群多样性, 而种群多样性的增加换句话说也可以说成是对可能出现的过度开发情形的避免或一种改善。

3.3 CDL-MCP SO 不同迭代时刻寻优能力的变化以及勘探和开发能力的平衡性

算法迭代前期, 将种群划分为多个子群并采用拉丁超立方体采样方法对种群进行初始化, 保证了种群粒子在搜索空间中是均匀分布的, 多样性好, 算法具有很好的勘探能力, 迭代过程中主群执行策略 CLS 的学习样本由于是粒子的个体最优通过向所有其他粒子的个体最优进行概率性学习, 综合了所有其他粒子到目前为止找到的优异信息, 从群的速度更新公式中虽然加入了个体最优向全局最优的学习项, 但是在算法在前期迭代过程中, 从群学习样本 X_i^d 对粒子的指引作用相比 X_i^g 以及主群着重于勘探的学习策略在前期迭代过程中发挥不明显, 能够保证种群可以在搜索空间进行充分地勘探。

算法迭代中期, 此时种群中每个粒子找到的个体最优值相比迭代前期已经有了很大的改善, 这是由于迭代前期种群优秀的勘探能力所发挥的作用而产生的结果, 此时不论是对于主群还是从群学习样本的产生, 由于被学习对象的解变优, 使得学习样本的值也随之变好, 整个种群的更新都在朝最终的最优解慢慢靠近, 种群勘探和开发能力并存, 优化了解的质量但不会使粒子过快收敛, 也保留了部分勘探能力能够继续探索可能的较优解。

算法迭代后期, 随着种群搜索到的解质量的进一步提高, 从群中学习样本 X_i^d 的值越来越接近直至等于全局最优, 而另外的学习样本 X_i^g 即使是向其他粒子的个体最优进行学习所得, 但由于迭代过程已经到达整个迭代周期的后期, 此时每个粒子找到的个体最优解的质量已经离问题的最终解很近了, 此时 X_i^g 的值浮动逐渐减小, 此时种群不再进行大范围的探索, 主要围绕在当前找到的最优解附近进行优化, 因为前期良好的勘探能力和中期适应的勘探能力使得粒子对问题空间已经进行了高度地探索, 所以在后期粒子靠近全局最优的过程中, 是不存在靠近的是局部最优的情形, 因此在后期整个种群主要发挥的是开发能力。

综上, 改进粒子群算法的寻优能力能够根据算法不同迭代时刻的不同需求分别发挥勘探或开发能力或同时发挥勘探和开发能力, 实现了两者之间的有效平衡。

3.4 CDL-MCP SO 算法的收敛性分析

Solis 等人^[25]的研究结果证明了随机优化算法以概率 1 收敛于全局最优解的条件, 主要有以下结论:

假设 1 若 $f(D(x, \zeta)) \leq f(x)$, 并且如果 $\zeta \in S$, 则有:

$$f(D(x, \zeta)) \leq f(\zeta)$$

其中, f 为目标函数, D 为产生问题解的函数, x 为 S

中搜索到的一个点, ζ 为从概率空间 (R^n, B, μ_ζ) 产生的随机向量, S 为问题的约束空间。

假设 2 S 的任意 Borel 子集 A , 若其测度 $\nu(A) > 0$, 则有:

$$\prod_{t=0}^{\infty} (1 - \mu_t(A)) = 0$$

其中, $\nu(A) > 0$ 为子集 A 的 n 维闭包, $\mu_t(A)$ 为由测度 μ_t 产生 A 的概率。

定理 1 假设目标函数 f 为可测函数, 问题的约束空间 S 为可测子集, $\{x_t\}_{t=0}^{\infty}$ 为随机算法产生的解序列, 则当满足假设 1 和假设 2 时, 有:

$$\lim_{t \rightarrow \infty} P\{x_t \in R_\epsilon\} = 1$$

其中, R_ϵ 为全局最优点的集合。

因此, 对任一随机优化算法, 只要能够满足假设 1 与假设 2, 就可以保证它能以概率 1 收敛于全局最优解。

定理 2 CDL-MCP SO 算法能够以概率 1 收敛于全局最优解

证明 CDL-MCP SO 算法满足假设 1。

设 CDL-MCP SO 算法的迭代函数 D 可以表示为

$$D(p_{g,t}, x_{i,t}) = \begin{cases} p_{g,t}, f(p_{g,t}) \leq f(x_{i,t}) \\ x_{i,t}, f(x_{i,t}) \leq f(p_{g,t}) \end{cases}$$

容易证明 CDL-MCP SO 满足假设 1。

证明 CDL-MCP SO 算法满足假设 2。

设规模为 N 的粒子群的样本空间的并必须包含 S , 即 $S \subseteq \bigcup_{i=1}^N M_{i,t}$ 。其中, $M_{i,t}$ 为第 t 代粒子 i 的样本空间的支撑集。对于满足 $x_{i_0,t} = p_t = p_g$ 的粒子 i_0 , $M_{i_0,t} = S$, 其他粒子 i 则满足:

对标准 PSO 算法, 设 $\varphi_1 = c_1 r_1$, $\varphi_2 = c_2 r_2$, 将速度更新公式代入位置更新公式后, 代入 $M_{i,t}$ 可得:

$$M_{i,t} = x_{i,j,t-1} + \omega(x_{i,j,t-1} - x_{i,j,t-2}) + \varphi_1(p_{i,j,t-1} - x_{i,j,t-1}) + \varphi_2(p_{g,j,t-1} - x_{i,j,t-1})$$

其中, $0 \leq \varphi_1 \leq c_1$, $0 \leq \varphi_2 \leq c_2$ 。显然, $M_{i,t}$ 表示一个由 φ_1 、 φ_2 所确定的超矩形, 其中一个端点为 $\varphi_1 = \varphi_2 = 0$, 另一个为 $\varphi_1 = c_1$, $\varphi_2 = c_2$ 。

当 $\max(c_1 |p_{i,j,t-1} - x_{i,j,t-1}|, c_2 |p_{g,j,t-1} - x_{i,j,t-1}|) < 0.5 \times \text{diam}_j(S)$ 成立时, 显然有 $\nu(M_{i,t} \cap S) < \nu(S)$ 。其中, $\text{diam}_j(S)$ 表示 S 在第 j 维分量的长度。由于 $x_t \rightarrow (c_1 p_t + c_2 p_g) / (c_1 + c_2)$, 因而 $\lim_{t \rightarrow \infty} M_{i,t} = 0$, 从而, 随迭代次数 t 的增加, $\nu(M_{i,t})$ 不断减少, 其并 $\bigcup_{t=0}^{\infty} M_{i,t}$ 也在减少。从而 $\nu(\bigcup_{t=0}^{\infty} M_{i,t} \cap S) < \nu(S)$, 表示存在整数 t' , 使得当 $t > t'$, 存在集合 $A \subseteq S$, 使得 $\sum_{t=0}^{\infty} \mu_{i,t}(A) = 0$, 这表明标准 PSO 算法不满足假设 2, 不能以概率 1 收敛于全局最优解。

而 CDL-MCP SO 算法是在标准 PSO 算法的基础上, 运用多子群、不同学习策略以及种群之间的协作交换机制, 产生新的粒子。因此, 对正常进化的粒子, 设其支撑集的并集为 α ; 利用以上改进策略产生的粒子, 设其支撑集的并集为 β 。

由于改进策略的指导性, 必然存在整数 t_1 , 使得当 $t > t_1$ 时, $\beta \supseteq S$ 。因此对于 CDL-MCP SO 算法, 存在整数 t_2 , 使得当 $t > t_2$ 时, $\alpha \cup \beta \supseteq S$ 。

定义 S 的任意 Borel 子集 $A = M_{i,t}$, 则有 $\nu(A) > 0$, $\mu_t(A) = \sum_{i=1}^N \mu_{i,t}(A) = 1$, 即 $\prod_{t=0}^{\infty} (1 - \mu_t(A)) = 0$, 所以 CDL-MCP SO 算法满足假设 2。

结论 1 CDL-MCP SO 算法是全局收敛算法。

证明 设 $\{p_{g,t}\}_{t=0}^{\infty}$ 为 CDL-MCP SO 算法生成的解序列, 由于 CDL-MCP SO 算法满足假设 1 和假设 2, 通过定理可知: $\lim_{t \rightarrow \infty} P\{p_{g,t} \in R_\epsilon\} = 1$ 成立, 故 CDL-MCP SO 算法能够以概率 1 收敛于全局最优解。

算法的收敛速度和解的精度在实验部分进行了详细分析, 见后文。

3.5 算法的时间复杂度分析

传统 PSO 算法主要包括以下几个部分: 初始化、适应度值的评估以及速度和位置更新, 他们的复杂度分别为

$O(2 \times N \times D)$ 、 $O(N \times D)$ 和 $O(2 \times N \times D)$ (N 和 D 分别表示种群规模和维度。因此, PSO 的时间复杂度为 $O(N \times D)$ 。与传统 PSO 相比, CDL-MCP SO 算法需要构建学习样本、主从群信息交流时评估所有从群本地最优。由于主从群只有在连续迭代次数达到刷新间隙时才进行信息交流, 并且评估的本地最优分别来自四个从群, 每次交流只评估四组数据, 因此多群协作过程的时间复杂度为 $O(3)$, 并且只有当粒子的个人最佳位置更新或个人最佳位置在几代内没有更新时, 才需要重建学习样本。构建学习样本的最坏情况时间复杂度是 $O(N \times D)$, 判断迭代是否达到刷新间隙以及是否停止的时间复杂度为 $O(2)$, 因此, CDL-MCP SO 的最坏情况时间复杂度也是 $O(N \times D)$, 包括初始化 $O(2 \times N \times D)$, 评估 $O(N \times D + 5)$, 更新 $O(2 \times N \times D + N \times D)$ 。根据以上分析, 可以得知 CDL-MCP SO 算法的时间复杂度与基本 PSO 算法处于同一水平。

4 实验验证与分析

4.1 典型迭代时刻的个体分布图及种群多样性的分析

给出了种群分别在 50、500 和 900 代时的个体分布图, 如图 4~6 所示。

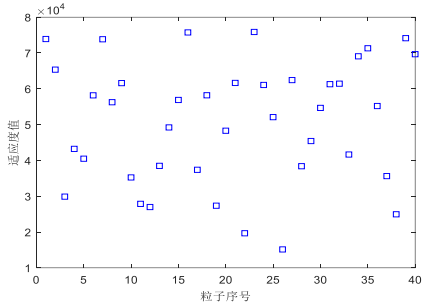


图 4 种群在 50 代时的个体分布图

Fig. 4 Individual distribution of the population at the 50th generation

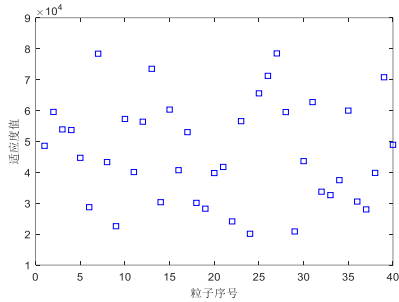


图 5 种群在 500 代时的个体分布图

Fig. 5 Individual distribution of the population at the 500th generation

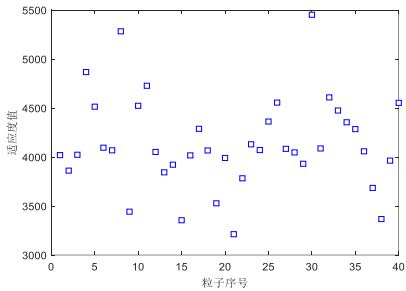


图 6 种群在 900 代时的个体分布图

Fig. 6 Individual distribution of the population at the 900th generation

图 4 代表种群在迭代前期的大致分布, 主从群独立执行 CLS 和 DLS, CLS 学习对象为所有其他粒子的个体最优, DLS 公式第二项学习对象虽为全局最优, 但得到的样本距离成为全局最优还要些许迭代时间, 种群多样性良好, 粒子基本均匀散落在搜索空间。

图 5 代表种群在迭代中期的大致分布, 从群迭代 DLS 公

式第二项得到的样本已经接近全局最优, 但主群中其他粒子个体最优的差异仍然保持学习后的样本具备较好的多样性, DLS 的一部分开发能力被中和, 加上种群每隔一定迭代次数就会进行解信息的交换, 粒子只是慢慢开始出现收敛的趋势, 但仍具备较好的种群多样性。

图 6 代表种群在迭代后期的大致分布, 从群迭代 DLS 公式第二项的学习样本等于全局最优, 后两项均发挥开发作用, 种群多样性减弱, 大部分粒子聚集在全局最优解附近, 但 CLS 的存在, 仍然会使个别粒子散落在距离全局最优解不太远的位置。

4.2 测试函数

为验证 CDL-MCP SO 性能, 采用 Benchmark 测试函数集中的 10 个函数进行求解验证, 其中包括两个单峰函数 f_1 - f_2 、五个多峰函数 f_3 - f_7 、三个组合函数 f_8 - f_{10} , 测试函数的具体信息如表 3 所示。

表 3 实验使用的 10 个测试函数

Tab. 3 The 10 test functions used in the experiment			
	No.	Functions	$F_i^* = F_i(x^*)$
Unimodal Functions	f_1	Rotated High Conditioned Elliptic Function	-1300
	f_2	Rotated Discus Function	-1100
	f_3	Rotated Weierstrass Function	-600
Basic Multimodal Functions	f_4	Rotated Griewanks Function	-500
	f_5	Rotated Katsuura Function	200
	f_6	Lunacek Bi_Rastrigin Function	300
	f_7	Rotated Lunacek Bi_Rastrigin Function	400
Composition Functions	f_8	Composition Function 2 (n=3,Unrotated)	800
	f_9	Composition Function 3 (n=3,Rotated)	900
	f_{10}	Composition Function 4 (n=3,Rotated)	1000

4.3 对比算法

为了验证 CDL-MCP SO 算法在求解复杂问题时的性能, 将其与统一粒子群算法(unified particle swarm optimization, UPSO)^[13]、完全知情粒子群算法(fully informed particle swarm, FIPS)^[11]维度学习粒子群算法(particle swarm optimization based on dimensional learning strategy, DLPSO)^[21]、并行综合学习粒子群优化算法(parallel comprehensive learning particle swarm optimization, PCLPSO)^[17]及标准粒子群算法(A modified particle swarm optimizer, PSO-W)^[2]五种改进粒子群算法进行对比实验。为了保证测试的公平性, 算法的参数设置均相同: 种群规模为 40, 迭代次数为 1000, 所有算法分别在 $D=10$ 、30、50 三个维度下在每个测试函数上独立运行 30 次。实验环境设置如下: AMD Ryzen 5350H with Radeon Vega Mobile Gfx 2.10 GHz, RAM 16GB, Windows 10 操作系统, Matlab R2016b。CDL-MCP SO 算法及对比算法的参数设置如表 4 所示。

表 4 CDL-MCP SO 算法及对比算法的参数设置

Tab. 4 CDL-MCP SO algorithm and comparison algorithm parameter setting		
算法	参数设置	参考文献
PSO-W	$\omega=0.9-0.4, c1=c2=2$	[2]
UPSO	$\chi: 0.729, c=1.49445$	[13]
FIPS	$\chi: 0.729, c=2$	[11]
DLPSO	$\omega=0.9-0.4, c1=1.5, c2=0.5-2.5$	[21]
PCLPSO	$\omega=0.9-0.4, c=1.49445$	[17]
CDL-MCP SO	$\omega=0.9-0.4, c1=c2=2.05, c3=2.0, c=1.49445$	—

4.4 算法性能分析

表 5~6 表示在 D=30,50 两种维度情况下, 每种算法在测试函数上的平均值和标准偏差。六种算法的最佳结果均以粗体显示出来。从表格数据可以看出 CDL-MCPSO 算法除了测试函数 f_2 上的性能略显不足以外, 能够在其他绝大多数不管是单峰

函数、多模态函数还是组合函数上均能找到更好的或相当的优化结果, 只在个别函数上稳定性稍显不足, 产生以上实验结果的主要原因是主从群采用的综合学习策略(CLS)和综合维度学习策略(CDL)在算法迭代前期发挥了优秀的勘探能力, 探索了可能存在较优解的所有区域, 为最终解质量的提高奠定了基础。

表 5 对比算法在测试函数上的优化结果(D=30)

Tab. 5 The optimization results of the comparison algorithm on the test function(D=30)

函数	Criterion	PSO-W	UPSO	FIPS	DLPSO	PCLPSO	CDL-MCPSO
f_1	Mean	1.07E+08	1.70E+07	1.67E+08	5.36E+07	3.26E+08	1.60E+07
	Std	8.06E+07	1.38E+07	1.86E+07	6.23E+07	8.16E+07	4.55E+06
f_2	Mean	1.04E+05	5.45E+04	8.45E+04	6.27E+04	3.95E+04	4.79E+04
	Std	3.01E+04	1.67E+04	2.22E+04	2.04E+04	2.26E+04	4.10E+04
f_3	Mean	2.52E+02	2.47E+02	3.93E+01	3.12E+01	3.06E+01	2.39E+01
	Std	1.51E+00	3.25E+02	3.78E+00	1.74E+00	3.35E+00	3.22E+01
f_4	Mean	2.42E+03	4.08E+03	5.78E+03	4.74E+03	4.24E+03	3.76E+03
	Std	3.42E+03	2.24E+03	7.14E+02	3.05E+02	6.18E+02	7.50E+02
f_5	Mean	2.28E+02	2.28E+02	3.09E+00	2.47E+00	2.32E+00	1.33E+00
	Std	3.38E+02	3.39E+02	3.01E-01	4.64E-01	5.96E-01	4.39E-01
f_6	Mean	6.92E+02	3.32E+02	2.35E+02	2.92E+02	1.93E+02	9.32E+01
	Std	1.26E+02	2.68E+02	2.74E+01	6.07E+01	7.38E+01	1.12E+01
f_7	Mean	6.84E+02	3.83E+02	2.46E+02	2.61E+02	2.39E+02	1.34E+02
	Std	1.19E+02	2.34E+02	1.91E+01	5.46E+01	1.05E+02	1.46E+01
f_8	Mean	5.51E+03	3.25E+03	6.74E+03	5.89E+03	3.58E+03	4.29E+02
	Std	3.49E+03	1.89E+03	2.19E+02	1.56E+03	8.48E+02	3.85E+02
f_9	Mean	8.48E+03	7.77E+03	8.97E+03	7.95E+03	6.61E+03	5.30E+03
	Std	3.47E+03	2.57E+03	2.52E+02	8.83E+02	8.85E+02	2.66E+02
f_{10}	Mean	4.31E+02	4.21E+02	3.03E+02	2.88E+02	2.94E+02	2.15E+02
	Std	3.68E+00	2.10E+02	2.04E+02	1.08E+01	7.63E+00	5.01E+00

表 6 对比算法在测试函数上的优化结果(D=50)

Tab. 6 The optimization results of the comparison algorithm on the test function(D=50)

函数	Criterion	PSO-W	UPSO	FIPS	DLPSO	PCLPSO	CDL-MCPSO
f_1	Mean	6.63E+08	4.03E+08	3.64E+08	2.12E+08	2.27E+08	2.60E+07
	Std	2.68E+08	3.10E+07	5.13E+07	3.74E+07	1.64E+08	1.05E+07
f_2	Mean	9.46E+04	1.63E+05	1.72E+05	9.73E+04	8.79E+04	9.93E+04
	Std	7.02E+04	6.54E+04	2.03E+04	2.62E+04	2.12E+04	3.42E+04
f_3	Mean	2.75E+02	2.67E+02	7.43E+01	5.76E+01	6.11E+01	4.90E+01
	Std	2.27E+00	3.12E+02	3.44E+00	3.60E+00	5.09E+00	2.06E+00
f_4	Mean	1.88E+04	2.17E+04	1.92E+04	1.71E+04	1.90E+04	1.52E+04
	Std	6.77E+03	4.36E+02	1.59E+03	1.10E+03	1.67E+03	4.94E+03
f_5	Mean	2.28E+02	2.28E+02	4.15E+00	3.71E+00	3.65E+00	2.21E+00
	Std	3.37E-01	3.38E+02	3.38E+02	5.89E-01	4.75E-01	1.06E+00
f_6	Mean	1.61E+03	4.54E+02	5.32E+02	6.87E+02	6.81E+02	2.32E+02
	Std	7.17E+02	1.92E+02	4.35E+01	1.05E+02	2.98E+02	2.35E+01
f_7	Mean	1.62E+03	5.47E+02	5.30E+02	7.20E+02	6.64E+02	3.76E+02
	Std	7.21E+02	1.39E+02	2.69E+01	1.11E+02	2.35E+02	9.88E+01
f_8	Mean	1.00E+04	6.80E+03	1.34E+04	1.17E+04	9.00E+03	5.70E+03
	Std	4.94E+02	4.47E+03	2.08E+03	1.74E+03	1.32E+03	4.75E+02
f_9	Mean	7.68E+03	1.87E+03	1.51E+03	3.23E+03	2.97E+03	1.40E+03
	Std	1.93E+03	1.43E+03	1.32E+03	1.75E+03	5.26E+03	1.36E+03
f_{10}	Mean	4.90E+02	4.74E+02	3.92E+02	3.72E+02	3.77E+02	3.39E+02
	Std	2.79E+00	1.77E+02	1.36E+01	1.31E+01	1.57E+01	1.68E+02

图 7~12 分别展示了 CDL-MCPSO 算法在 f_1 (单峰)、 f_4 (多模态)和 f_9 (组合)三个函数上 30、50 维下的收敛曲线图, 针对函数 f_1 的收敛曲线图, 从图 7 可以观察到改进算法明显优于 PSO-W、UPSO、FIPS, 与 PCLPSO 相比, 改进算法找到了全局最优

解, 而 PCLPSO 不仅没有找到全局最优解, 而且从 50 代左右就开始收敛, 收敛速度过于快, 图 10 和图 7 的情况大致类似。针对函数 f_4 的收敛曲线图, 尤其是图 11, CDL-MCPSO 性能明显优于其他对比算法, 从 f_9 的收敛曲线图可以直观地看出 CDL-

chinaXiv:202204.00060v1

MCPSO 不仅在找到的解上明显优于其他对比算法,而且收敛速度也快于其他算法,在迭代后期没有出现“振荡”现象,解决了“维度退化”问题,虽然 CDL-MCPSO 算法也存在不足,算法稳定性在个别测试函数上相比其他算法略显不足,但总的来说,CDL-MCPSO 算法与其他算法相比,在优化结果上都大有提升。

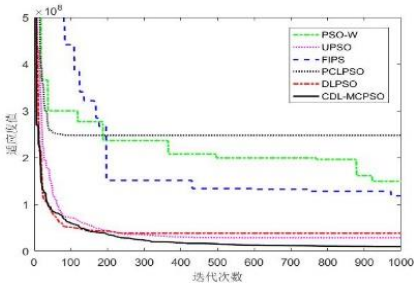


图 7 函数 f_1 的收敛曲线(D=30)
Fig. 7 Convergence curve of f_1 (D=30)

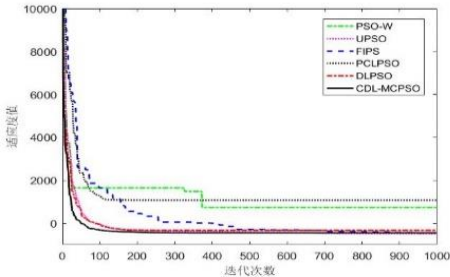


图 8 函数 f_4 的收敛曲线(D=30)
Fig. 8 Convergence curve of f_4 (D=30)

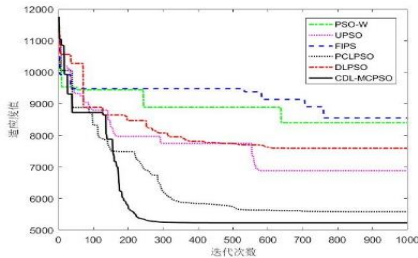


图 9 函数 f_9 的收敛曲线(D=30)
Fig. 9 Convergence curve of f_9 (D=30)

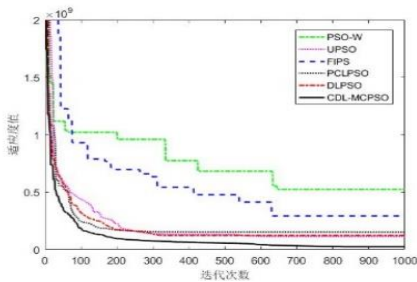


图 10 函数 f_1 的收敛曲线(D=50)
Fig. 10 Convergence curve of f_1 (D=50)

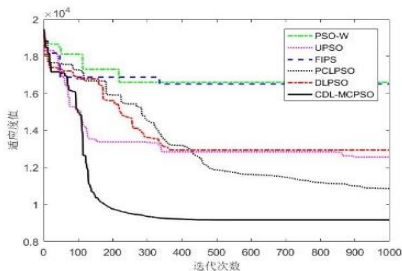


图 11 函数 f_4 的收敛曲线(D=50)
Fig. 11 Convergence curve of f_4 (D=50)

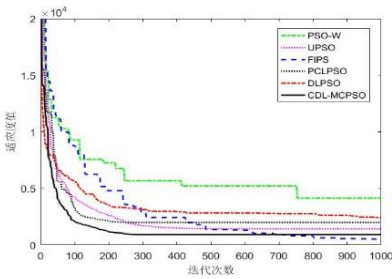


图 12 函数 f_9 的收敛曲线(D=50)
Fig. 12 Convergence curve of f_9 (D=50)

为了进一步验证 CDL-MCPSO 算法的性能,借鉴数据统计与分析方法,采用显著性水平为 0.05 的 Wilcoxon 秩检验方法来判断算法性能。其中“+”“—”“ \approx ”分别表示 CDL-MCPSO 算法的结果优于、劣于、相当于对应算法的测试结果。从表 7 的 Wilcoxon 测试结果来看,在 $\alpha=0.05$ 时 CDL-MCPSO 算法在测试函数上相较于对比算法均有一定优势。

表 7 通过 Wilcoxon 的测试得到结果

Tab. 7 Results obtained through Wilcoxon's test						
D	CDL-MCPSO VS	P-value	+	\approx	—	$\alpha=0.05$
30	PSO-W	0.000851	37	1	0	Yes
	UPSO	0.000112	36	1	0	Yes
	FIPS	0.000401	36	1	0	Yes
	DLPSO	0.000102	35	2	0	Yes
	PCLPSO	0.000099	36	1	0	Yes
50	PSO-W	0.000134	36	1	0	Yes
	UPSO	0.000026	35	1	0	Yes
	FIPS	0.000147	37	1	0	Yes
	DLPSO	0.000082	36	2	0	Yes
	PCLPSO	0.000096	36	1	0	Yes

根据以上对比分析,CDL-MCPSO 可以保持更好的多样性开发原始空间来获得高质量的解,并且具备最佳的达到最优的能力使算法快速收敛。因此,CDL-MCPSO 是一种能够有效提升 PSO 性能的可行方法。

5 结束语

为提高算法运行效率、平衡算法的勘探和开发能力,本文基于主从范式集群结构提出了综合维度学习的多群协作粒子群优化算法(CDL-MCPSO),主从群独立执行具有不同职能的学习策略,其中从群学习策略是针对 DLS 存在的“过度开发”问题而开发,同时为提升解的质量,还提出了一种新的解交换机制,用于群体之间的信息共享,实验结果证明 CDL-MCPSO 算法不仅能够绝大多数单峰、多峰和组合函数上找到更好的或相同的优化结果,而且在收敛速度和解的质量上均有明显优势,不仅解决了“振荡”和“维度退化”问题,还有效避免了“过度开发”。下一阶段工作主要是提高算法在部分测试函数上的稳定性并将其应用于实际复杂问题中进行有效性的验证。

参考文献:

[1] Eberhart R, Kennedy J. A new optimizer using particle swarm theory [C]// Proc of the 6th International Symposium on Micro Machine and Human Science. Japan: IEEE, 1995: 39-43.

[2] Shi Y, Eberhart R. A modified particle swarm optimizer [C]// IEEE International Conference on Evolutionary Computation Proceedings. Piscataway NJ: IEEE, 1998: 69-73.

[3] El Afia A, Sarhani M, Aoun O. Hidden markov model control of inertia weight adaptation for particle swarm optimization [J]. IFAC-PapersOnLine, 2017, 50 (1): 9997-10002.

- [4] Gou Jin, Lei Yuxiang, Guo Wangping, *et al.* A novel improved particle swarm optimization algorithm based on individual difference evolution [J]. *Applied Soft Computing*, 2017, 57: 468-481.
- [5] Nasir M, Das S, Maity D, *et al.* A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization [J]. *Information Sciences*, 2012, 209: 16-36.
- [6] 焦重阳, 周清雷, 张文宁. 混合拓扑结构的粒子群算法及其在测试数据生成中的应用研究 [J]. *计算机科学*, 2017, 44 (12): 255-260. (Jiao Chongyang, Zhou Qinglei, Zhang Wenning. Particle swarm optimization with mixed topology and its application in test data generation [J]. *Computer Science*, 2017, 44 (12): 255-260.)
- [7] Gong Yuejiao, Li Jingjing, Zhou Yicong, *et al.* Genetic learning particle swarm optimization [J]. *IEEE Trans on Cybernetics*, 2015, 46 (10): 2277-2290.
- [8] Aydılek I B. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems [J]. *Applied Soft Computing*, 2018, 66: 232-249.
- [9] Şenel F A, Gökçe F, Yüksel A S, *et al.* A novel hybrid PSO-GWO algorithm for optimization problems [J]. *Engineering with Computers*, 2019, 35 (4): 1359-1373.
- [10] 陈跃刚, 许奕. 基于综合学习策略的多目标分解粒子群算法 [J]. *微电子学与计算机*, 2018, 35 (10): 75-79. (Chen Yuegang, Xu Yi. Multi-objective decomposition particle swarm algorithm based on comprehensive learning strategy [J]. *Microelectronics and Computer*, 2018, 35 (10): 75-79.)
- [11] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler maybe better [J]. *IEEE Trans on Evolutionary Computation*, 2004, 8 (3): 204-210.
- [12] 田梦丹, 梁晓磊, 符修文, 等. 具有博弈概率选择的多子群粒子群算法 [J]. *计算机科学*, 2021, 48 (10): 67-76. (Tian Mingdan, Liang Xiaolei, Fu Xiuwen, *et al.* Multi-subgroup particle swarm optimization algorithm with game probability selection [J]. *Computer Science*, 2021, 48 (10): 67-76.)
- [13] Parsopoulos K E, Vrahatis M N. UPSO: A unified particle swarm optimization scheme [C]// *International Conference of Computational Methods in Sciences and Engineering*. Greece Florida: CRC Press, 2019: 868-873.
- [14] 袁小平, 蒋硕. 基于分层自主学习的改进粒子群优化算法 [J]. *计算机应用*, 2019, 39 (01): 148-153. (Yuan Xiaoping, Jiang Shuo. Improved particle swarm optimization algorithm based on hierarchical autonomous learning [J]. *Computer Applications*, 2019, 39 (01): 148-153.)
- [15] Li Dongyang, Guo Weian, Lerch A, *et al.* An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization [J]. *Swarm and Evolutionary Computation*, 2021, 60: 100789.
- [16] Zhou Shangbo, Sha Long, Zhu Shufang, *et al.* Adaptive hierarchical update particle swarm optimization algorithm with a multi-choice comprehensive learning strategy [J]. *Applied Intelligence*, 2022, 52 (2): 1853-1877.
- [17] Gulcu Ş, Kodaz H. A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization [J]. *Engineering Applications of Artificial Intelligence*, 2015, 45: 33-45.
- [18] 廖玮霖, 程杉, 尚冬冬, 等. 多策略融合的粒子群优化算法 [J]. *计算机工程与应用*, 2021, 57 (01): 69-76. (Liao Weilin, Cheng Shan, Shang Dongdong, *et al.* Multi-strategy fusion particle swarm optimization algorithm [J]. *Computer Engineering and Applications*, 2021, 57 (01): 69-76.)
- [19] 纪伟, 李英梅, 季伟东, 等. 粒子置换的双种群综合学习 PSO 算法 [J]. *计算机科学与探索*, 2021, 15 (04): 766-776. (Ji Wei, Li Yingmei, Ji Weidong, *et al.* PSO algorithm of dual-population comprehensive learning based on particle replacement [J]. *Journal of Computer Science and Exploration*, 2021, 15 (04): 766-776.)
- [20] Xu Guiping, Cui Quanlong, Shi Xiaohu, *et al.* Particle swarm optimization based on dimensional learning strategy [J]. *Swarm and Evolutionary Computation*, 2019, 45: 33-51.
- [21] Liang J J, Qin A K, Suganthan P N, *et al.* Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. *IEEE Trans on Evolutionary Computation*, 2006, 10 (3): 281-295.
- [22] Parsopoulos K E, Vrahatis M N. On the computation of all global minimizers through particle swarm optimization [J]. *IEEE Trans on Evolutionary Computation*, 2004, 8 (3): 211-224.
- [23] Van Den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization [J]. *IEEE Trans on Evolutionary Computation*, 2004, 8 (3): 225-239.
- [24] McKay M D, Beckman R J, Conover W J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code [J]. *Technometrics*, 2000, 42 (1): 55-61.
- [25] Solis F J, Wets R J B. Minimization by random search techniques [J]. *Mathematics of Operations Research*, 1981, 6 (1): 19-30.